

Systematic Construction of Temporal Logics for Dynamical Systems via Coalgebra

Baltasar Trancón y Widemann

Computer Science, Technical University of Ilmenau
Bayreuth, DE

Baltasar.Trancon@tu-ilmenau.de

December 11, 2012

Abstract

Temporal logics are an obvious high-level descriptive companion formalism to dynamical systems which model behavior as deterministic evolution of state over time. A wide variety of distinct temporal logics applicable to dynamical systems exists, and each candidate has its own pragmatic justification. Here, a systematic approach to the construction of temporal logics for dynamical systems is proposed: Firstly, it is noted that dynamical systems can be seen as coalgebras in various ways. Secondly, a straightforward standard construction of modal logics out of coalgebras, namely Moss's coalgebraic logic, is applied. Lastly, the resulting systems are characterized with respect to the temporal properties they express.

1 Introduction

Dynamical systems are the classical constructive formalism for behaviour arising from the deterministic evolution of system state over time [1], dating back to the works of Newton and Laplace. Clearly *temporal logics*, with operators such as ‘next’, ‘always’, ‘eventually’ and ‘for-at-least’, constitute a companion descriptive formalism. However, the relation is not one-to-one: On one hand, there is a unifying theory underlying the various perspectives on dynamical systems as monoid actions, which uniformly covers discrete and continuous, as well as hybrid systems [5]. But on the other hand, the diversity of temporal logics in literature is immense, cf. [9], and the choice for a particular system is often justified by ad-hoc pragmatic arguments. The present article explores a systematic and fairly generic approach to the construction of temporal logics for dynamical systems, via the rather recent mathematical field of *universal coalgebra* which appears to be intimately connected to both dynamical systems [8] and modal logics [4]. A different approach also based on coalgebras and the Stone duality has been suggested [2] for constructing modal logics of *transition systems*, a close relative of dynamical systems in computer science.

The method outlined in the remainder of this article, while theoretically simple, touches on many different fields of mathematics: order theory, category theory, algebra, coalgebra, classical modal logics la Kripke, and coalgebraic logics la Moss [6]. Thus a significant proportion of the available space is dedicated to reviewing the relevant definitions and propositions from the respective standard literature. This review makes up the sections 2 and 3. The expert reader is encouraged to skip ahead: Section 4 ties up all the loose ends and gives a novel contribution. There a selection of obvious coalgebraic perspectives on dynamical systems is explored, and the respective logics entailed by applying Moss's construction are characterized.

2 Review: Classical Ingredients

This section reviews some basic definitions and propositions.

2.1 Order Relations

We assume that the reader is familiar with basic order-theoretic properties of binary relations, namely with *reflexive*, *transitive*, *symmetric* relations, and with *preorders*, *partial orders* and *equivalences*. We

give two additional related definitions that are not quite as universal:

Definition 1. Let X be a set. A binary relation $R \subseteq X^2$ is called *non-branching* if and only if $x R y$ and $x R z$ imply $y R z$ or $z R y$, and *linear* if and only if $x R y$ or $y R x$, respectively, for all $x, y, z \in X$.

2.2 Monoids

We assume that the reader is familiar with the notions of a *monoid* $\mathbb{M} = (M, 0, +)$, and of monoid *generators* and *cyclic* monoids. Every monoid induces an ordering relation.

Definition 2 (Monoid Order). Let $\mathbb{M} = (M, 0, +)$ be a monoid. For any elements $a, b \in M$, we write $a \leq_{\mathbb{M}} b$ if and only if there is some $c \in M$ such that $a + c = b$. We say that $a \leq_{\mathbb{M}} b$ *via* c . It follows directly from the monoid axioms that $\leq_{\mathbb{M}}$ is reflexive and transitive, hence a preorder. By extension, \mathbb{M} itself is called *symmetric/non-branching/linear* if and only if $\leq_{\mathbb{M}}$ is symmetric/non-branching/linear, respectively.

Note that being symmetric in this sense is different from being Abelian. In fact, symmetry characterizes a subclass of monoids, the groups.

Lemma 3 (Groups). *A monoid \mathbb{M} is a group if and only if it is symmetric. Every symmetric monoid is trivially linear, with the degenerate order $(\leq_{\mathbb{M}}) = M^2$, the full relation.*

2.3 Dynamical Systems

Definition 4 (Dynamical System). Let $\mathbb{T} = (T, 0, +)$ be a monoid called *time*. A *dynamical system* is an enriched structure $\mathbb{S} = (\mathbb{T}, S, \Phi)$ with a set S called *state space*, and a map $\Phi : S \times T \rightarrow S$ called *dynamics*, such that

$$\Phi(s, 0) = s \qquad \Phi(\Phi(s, t), u) = \Phi(s, t + u) \qquad (1)$$

In other words, Φ is a *right monoid action* of \mathbb{T} on S . \mathbb{S} is called *linear-time* if and only if \mathbb{T} is linear, otherwise *nonlinear-time*, and *invertible* if and only if \mathbb{T} is symmetric.

Corollary. *There are no invertible nonlinear-time dynamical systems.*

Dynamical systems are a fundamental model class of many natural and social sciences. In comparison with their younger counterpart in computer science, automata and transition systems, dynamical systems are typically

- behaviourally weaker – deterministic, non-pointed (without distinguished initial states) and total (without spontaneous termination), but
- structurally stronger – with additional features of time (density, completeness) and state space (topology, metric, differential geometry, measures).

Automata-like construction can be emulated by dynamical systems; see examples below.

Definition 5 (Step, Trajectory, Orbit). From the dynamics map we may derive three forms of auxiliary functions:

$$\begin{array}{lll} \Phi^t : S \rightarrow S & \Phi_s : T \rightarrow S & \Phi^\circ : S \rightarrow \mathcal{P}S \\ \Phi^t(s) = \Phi(s, t) & \Phi_s(t) = \Phi(s, t) & \Phi^\circ(s) = \text{Img}(\Phi_s) = \{\Phi(s, t) \mid t \in T\} \end{array}$$

Φ^t is called the *step of duration* t , or just the t -step. Φ_s is called the *trajectory of initial state* s . $\Phi^\circ(s)$ is called the *orbit* of state s .

Lemma 6 (Homomorphic Steps). *The dynamical systems with time \mathbb{T} are precisely those systems (\mathbb{T}, S, Φ) such that the step construction is a monoid homomorphism from \mathbb{T} into the monoid of functions of type $S \rightarrow S$ with right composition.*

$$\Phi^0 = \text{id}_S \qquad \Phi^{t+u} = \Phi^u \circ \Phi^t \qquad (2)$$

where $\text{id}_X(x) = x$ and $(f \circ g)(x) = f(g(x))$ for all x .

Corollary (Generating Steps). *If $G \subseteq T$ is a generator of \mathbb{T} , then Φ is determined uniquely by the collection of steps $\{\Phi^t \mid t \in G\}$.*

Example (Instances of Time).

- The time monoid $(\mathbb{N}, 0, +)$ yields standard non-invertible, discrete-time dynamical systems. The step Φ^1 is generating. Trajectories are (one-sided) infinite sequences.
- The time monoid $(\mathbb{Z}, 0, +)$ yields standard invertible, discrete-time dynamical systems. The step Φ^1 is generating and must be invertible. Trajectories are two-sided infinite sequences.
- The time monoid $(\mathbb{R}_+, 0, +)$ yields standard non-invertible, continuous-time dynamical systems. No simple step generator exists. Trajectories are one-sided parametric curves.
- The time monoid $(\mathbb{R}, 0, +)$ yields standard invertible, continuous-time dynamical systems. No simple step generator exists; classical definitions are given as solutions to ordinary differential equations. Trajectories are two-sided parametric curves.
- The “time” monoid $(\Sigma^*, \varepsilon, \cdot)$ over some finite alphabet Σ yields total semiautomata, or deterministic finitely-labelled transition systems. The steps $\{\Phi^a \mid a \in \Sigma\}$ (columns of the transition table) are generating. Trajectories are big-step transition functions of total automata, mapping input words to final states.

2.4 Propositional Modal Logics

We assume that the reader is familiar with the syntax and semantics of classical propositional logics and their presentation in terms of the connectives \neg and \rightarrow .

Definition 7 (Syntax of Propositional Modal Logics). The modal extension of classical propositional logics adds two unary connectives \Box and \Diamond , taking \Box as primitive and defining

$$\Diamond A = \neg \Box \neg A$$

Definition 8 (Semantics of Propositional Modal Logics). A *normal* modal extension of classical propositional logics adds at least the deduction rule of *necessitation* or *generalization*, and the axiom of *distribution*:

$$A \vdash \Box A \qquad \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$$

Example. Important normal modal logics are obtained by adding certain axioms:

- $\Box A \rightarrow A$ added to the minimal system results in the logic T .
- $\Box A \rightarrow \Box \Box A$ added to T results in the logic $S4$.
- $\Box(\Box A \rightarrow B) \vee \Box(\Box B \rightarrow A)$ added to $S4$ results in the logic $S4.3$.
- $\Diamond A \rightarrow \Box \Diamond A$ added to $S4$ or $S4.3$ results in the logic $S5$.

2.5 Kripke Semantics

Definition 9 (Kripke Frame). A Kripke frame is a structure (W, R) with a set W of *worlds* and a relation R on W called *accessibility*.

Definition 10 (Kripke Model). Let (W, R) be a Kripke frame. A Kripke model (of propositional modal logic) is an extended structure (W, R, \Vdash) , where \Vdash is a relation between W and the language *Prop* of logical formulas, such that

$$\begin{aligned} w \Vdash \neg A & \iff w \not\Vdash A \\ w \Vdash A \rightarrow B & \iff w \not\Vdash A \text{ or } w \Vdash B \\ w \Vdash \Box A & \iff v \Vdash A \text{ whenever } w R v \end{aligned}$$

We say that w *satisfies* A in (W, R, \Vdash) if and only if $w \Vdash A$.

Lemma 11. *The satisfaction relation \Vdash of a Kripke frame is determined uniquely by the satisfaction of atomic propositions.*

Definition 12 (Validity). A formula A is called *valid* in

- a world w if and only if w satisfies A ,
- a Kripke model (W, R, \Vdash) if and only if it is valid in all worlds $w \in W$,
- a Kripke frame (W, R) if and only if it is valid in all Kripke models (W, R, \Vdash) ,
- a class C of Kripke frames if and only if it is valid in all members of C .

Definition 13 (Soundness/Completeness). A propositional modal logic L is called, with respect to a class C of Kripke frames, *sound* if and only if truth in L implies validity in C , and *complete* if and only if validity in C implies truth in L .

Theorem 14 (Soundness/Completeness). *The modal logics $S4/S4.3/S5$ are sound and complete for the class of Kripke frames (W, R) where R is an arbitrary/non-branching/symmetric preorder, respectively.*

Definition 15 (Finite Model Property). A propositional modal logic L is said to have the *finite model property*, if and only if it is complete for a class of finite Kripke frames.

Theorem 16. *The modal logics $S4/S4.3/S5$ have the finite model property, for subclasses of the respective classes given in Theorem 14.*

3 Review: Additional Ingredients

This section reviews some definitions and propositions that are also basic, but from less well-known fields. See [8, 6] for greater detail.

3.1 Category Theory

Definition 17 (Set Endofunctor). A *functor* F on the category of sets, or *set endofunctor*, is a map that assigns to every set X a set FX , and to every function $h : X \rightarrow Y$ a function $Fh : FX \rightarrow FY$, such that

$$F(\text{id}_X) = \text{id}_{FX} \qquad F(g \circ h) = Fg \circ Fh$$

where $\text{id}_X(x) = x$ and $(g \circ f)(x) = g(f(x))$.

All functors considered in the following are tacitly set endofunctors.

Definition 18 (Monotonic Functor). A functor F is called *monotonic* if and only if $X \subseteq Y$ implies $FX \subseteq FY$.

Coalgebraic logics deal with a class of functors called *standard*, which are essentially monotonic, plus an additional condition, namely preservation of weak pullbacks, that is rather technical but fortunately inessential for the present discussion.

Definition 19 (Finitary Functor). A functor is called *finitary* if and only if

$$FX \subseteq \bigcup \{FY \mid Y \subseteq X; Y \text{ finite}\}$$

otherwise *infinitary*. For monotonic finitary functors, the above is necessarily an equality. A standard, infinitary functor F has a *finitary restriction* F_f defined by

$$F_f X = \bigcup \{FY \mid Y \subseteq X; Y \text{ finite}\} \qquad F_f(h : X \rightarrow Y) = Fh|_{F_f X}$$

Definition 20 (Functor Product). The pointwise Cartesian product of functors is again a functor.

$$(F \times G)X = FX \times GX \qquad ((F \times G)h)(x, y) = ((Fh)(x), (Gh)(y))$$

Example. The following are standard functors:

- The *identical* functor \mathcal{I}

$$\mathcal{I}X = X \qquad \mathcal{I}h = h$$

\mathcal{I} is finitary; hence $\mathcal{I}_f = \mathcal{I}$.

- The *constant* functor $_@C$ for some set C

$$X@C = C \qquad h@C = \text{id}_C$$

$_@C$ is finitary.

- the *powerset* functor \mathcal{P}

$$\mathcal{P}X = \{W \mid W \subseteq X\} \qquad (\mathcal{P}h)(W) = \{h(x) \mid x \in W\}$$

\mathcal{P} is not finitary; its finitary restriction is the *finite powerset* functor \mathcal{P}_f .

- the *Hom* functor $_^C$ for some set C

$$X^C = \{f \mid f : C \rightarrow X\} \qquad (h^C)(g) = h \circ g$$

$_^C$ is finitary if and only if C is finite; its finitary restriction is the *image-finite* functor $_f^C$.

Clearly, a relation $R \in \mathcal{P}(X \times Y)$ is precisely the set of pairs (x, y) for which there is some $r \in R$ such that $\pi_1(r) = x$ and $\pi_2(r) = y$, where π_1, π_2 are the natural projections from the binary Cartesian product. This seemingly redundant presentation suggests an interaction of relations and functors.

Definition 21 (Relational Lifting). Let F be a functor. Every relation $R \in \mathcal{P}(X \times Y)$ has a *lifting* $F[R] \in \mathcal{P}(FX \times FY)$ defined as the set of pairs (\hat{x}, \hat{y}) for which there is some $\hat{r} \in FR$ such that $(F\pi_1)(\hat{r}) = \hat{x}$ and $(F\pi_2)(\hat{r}) = \hat{y}$.

Example. The liftings for the functors discussed above are as follows:

- The identical functor lift a relation to itself: $x \mathcal{I}[R] y$ if and only if $x R y$.
- The constant functor lifts to the identity relation: $c [R]@C c'$ if and only if $c = c' \in C$.
- $Y \mathcal{P}[R] Z$ if and only if for all $y \in Y$ there is a $z \in Z$, and vice versa, such that $y R z$.
- $f [R]^C g$ if and only if $f(c) R g(c)$ for all $c \in C$.

3.2 Universal Coalgebra

Definition 22 (Coalgebra). Let F be a functor. An F -coalgebra is a structure (X, f) with an object X and an arrow $f : X \rightarrow FX$.

Definition 23 (Homomorphism). Let F be a functor. Let (X, f) and (Y, g) be F -coalgebras. An F -coalgebra homomorphism from (X, f) to (Y, g) is an arrow $h : X \rightarrow Y$ such that $Fh \circ f = g \circ h$. We write $h : (X, f) \rightarrow (Y, g)$ or simply $h : f \rightarrow g$.

Definition 24 (Final Coalgebra). Let F be a functor. An F -coalgebra (Z, z) is called *final* if and only if there is a unique homomorphism $f! : f \rightarrow z$ from any other F -coalgebra.

Theorem 25. *Every finitary functor has a final coalgebra.*

Definition 26 (Bisimulation). Let F be a functor. Let (X, f) and (Y, g) be F -coalgebras. A *bisimulation* between (X, f) and (Y, g) is a relation $R \subseteq X \times Y$ that can be extended to an F -coalgebra (R, r) such that the projections are coalgebra homomorphisms $\pi_1 : r \rightarrow f$ and $\pi_2 : r \rightarrow g$. We say that states $x \in X$ and $y \in Y$ are *bisimilar* if and only if there is a bisimulation relating them.

The final coalgebra can be seen as a system of representatives of equivalence classes modulo bisimilarity.

Theorem 27. *Let F be a standard functor. If a final F -coalgebra (Z, z) exists then, for given F -coalgebras (X, f) and (Y, g) , two states $x \in X; y \in Y$ are bisimilar if and only if $f!(x) = g!(y)$.*

Definition 28 (Parallel Coalgebra Composition). Coalgebras with the same carrier can be combined in parallel: Let (X, f) be an F -coalgebra and (X, g) be a G -coalgebra. Then $(X, \langle f, g \rangle)$ is an $(F \times G)$ -coalgebra, where

$$\langle f, g \rangle(x) = (f(x), g(x))$$

3.3 Moss's Coalgebraic Logic

The idea of Moss's coalgebraic logic [6] is to replace Kripke frames by F -coalgebras for some functor F , and to derive a universal and natural modality from F itself.

Definition 29 (Moss's Coalgebraic Logic, Abstract). Fix a standard functor F . Extend the syntax of propositional logic by a pseudo-unary connective ∇ that, unlike the classical modalities like \Box , applies not to a single formula $A \in \text{Prop}$ but to an expression of type either $\hat{A} \in F(\text{Prop})$ or $\hat{A} \in F_i(\text{Prop})$. For infinitary F where the choice makes a difference, the cases are called *infinitary* and *finitary* F -coalgebraic logics, respectively. A Moss model is a structure (X, f, \Vdash) where (X, f) is an F -coalgebra and \Vdash is a relation between coalgebra states and formulas, such that

$$x \Vdash \neg A \iff x \not\Vdash A \qquad x \Vdash A \rightarrow B \iff x \not\Vdash A \text{ or } x \Vdash B$$

as for Kripke models, but

$$x \Vdash \nabla \hat{A} \iff f(x) F[\Vdash] \hat{A}$$

Moss's coalgebraic logic as presented here specifies satisfaction only up to atomic propositions, in analogy to Kripke frames. In Moss's original presentation, the specification is unique, in analogy to Kripke models.

Definition 30 (Moss's Coalgebraic Logic, Concrete). Let (X, f) be an F -coalgebra. Let $s : X \rightarrow \mathcal{P}(\text{Prop}_0)$ be the map that assigns to each state $x \in X$ the desired set of valid atomic propositions. Then (X, s) is a $\text{Const}(\mathcal{P}(\text{Prop}_0))$ -coalgebra. For the parallel composite coalgebra $(X, g = \langle f, s \rangle)$, a unique Moss model is specified by the additional clause

$$x \Vdash A \iff A \in s(x) \quad (A \in \text{Prop}_0)$$

The following two propositions state that traditional Kripke frames are essentially equivalent to the special case $F = \mathcal{P}$.

Lemma 31. \mathcal{P} -coalgebras (X, f) are in one-to-one correspondence to relations R on X by putting $x R y$ if and only if $y \in f(x)$.

Theorem 32. The Kripke modalities \Box, \Diamond and the Moss modality ∇ for finitary \mathcal{P} -coalgebraic logics are equivalent. For infinitary \mathcal{P} -coalgebraic logics, they are also equivalent in the presence of infinitary conjunction and disjunction; otherwise ∇ is generally more expressive.

$$\begin{aligned} w \Vdash_K \Box A &\iff w \Vdash_M \nabla \{A\} \vee \nabla \emptyset & w \Vdash_K \Diamond A &\iff w \Vdash_M \nabla \{A, \top\} \\ w \Vdash_M \nabla \hat{A} &\iff w \Vdash_K \Box \left(\bigvee \hat{A} \right) \wedge \Diamond \hat{A} & \text{where } \Diamond \hat{A} &= \{ \Diamond B \mid B \in \hat{A} \} \end{aligned}$$

where \Vdash_K / \Vdash_M denote satisfaction à la Kripke/Moss, respectively.

In general, the infinitary version of the operator ∇ is better matched with a logic where conjunction and disjunction are also infinitary. While an uncommon topic classically, infinitary logics are an important topic in modal logic because of their connection to bisimulation. The following theorem generalizes a theorem of Kripke-style logic, where bisimilarity is defined ad-hoc but equivalently to the coalgebraic notion specialized as in Lemma 31.

Theorem 33. In fully (\wedge, \vee, ∇) infinitary F -coalgebraic logic, two states $s, t \in S$ satisfy the same set of formulas if and only if they are bisimilar.

4 Constructions

This section gives novel theoretical results by investigating the ramifications of the following recipe:

1. identify some generic F -coalgebraic view on dynamical systems,
2. use Moss's construction to obtain logics with ∇_F modality, depending on the functor F ,
3. relate ∇_F to established temporal logic operators.

Note that all of the following constructions have the state space S of a fixed dynamical system as the carrier of some coalgebra for various functors. Hence the associated logical languages can coexist naturally in a single system, by the parallel composition given in Definition 28.

4.1 Step Logics

Definition 34 (Step Coalgebra). Let $\mathbb{S} = (\mathbb{T}, S, \Phi)$ be a dynamical system. For any element $t \in T$, the \mathcal{I} -coalgebra (S, Φ^t) is called the *t-step coalgebra* of \mathbb{S} .

Definition 35 (Multi-Step Coalgebra). Let $\mathbb{S} = (\mathbb{T}, S, \Phi)$ be a dynamical system. For any subset $U \subseteq T$, the \mathcal{I}^U -coalgebra $(S, s \mapsto \Phi_s \circ \text{in})$, given the inclusion map $\text{in} : U \rightarrow T$, is called the *U-multi-step coalgebra* of \mathbb{S} .

Lemma 36. *The ∇ modality of step coalgebras amounts to*

- for the *t-step*:

$$s \Vdash \nabla A \iff \Phi(s, t) \Vdash A$$

- for the *U-multi-step*:

$$s \Vdash \nabla \hat{A} \iff \Phi(s, t) \Vdash \hat{A}(t) \text{ for all } t \in U$$

The functors for t-steps and finite U-multi-steps are finitary; hence no additional distinction between finitary and infinitary logics arises.

Definition 37 (Step Modality).

$$\bigcirc A = \nabla A \qquad \bigcirc_t A = \nabla u \mapsto \begin{cases} A & (t = u) \\ \top & (t \neq u) \end{cases}$$

Example. (Multi-)Step coalgebras are of particular interest for finite generators, since they specify the dynamics uniquely and concisely. The following are generating, cf. Example 2.3:

- For time $(\mathbb{N}, 0, +)$, the 1-step coalgebra maps every state to its successor. The resulting temporal logic has \bigcirc as the *next* operator of traditional unidirectional discrete-time temporal logic.
- For time $(\mathbb{Z}, 0, +)$, the (± 1) -step coalgebra maps every state to its successor/predecessor, respectively. The resulting temporal logic has $\bigcirc_{\pm 1}$ as the *next/previous* operators of traditional bidirectional discrete-time temporal logic, respectively.
- For “time” $(\Sigma^*, \varepsilon, \cdot)$, the Σ -multi-step coalgebra maps every automaton state to its response function (row of the transition table). The resulting logic has $(\bigcirc_a)_{a \in \Sigma}$ as the generating cases of Pratt’s *necessity* operators $[a]$ in dynamic logic [7], where they are extended to the free Kleene algebra over Σ .

Interesting infinite, non-generating examples include:

- For time $(\mathbb{R}, 0, +)$ and $\delta > 0$, let U denote the open interval $(-\delta, \delta)$. The U -multi-step coalgebra maps every state to its temporal δ -neighbourhood.

Lemma 38. *The modality ∇ and the family of modalities $(\bigcirc_t)_{t \in U}$ for generating U are straightforwardly equivalent if U is finite, and equivalent in the presence of infinitary conjunction otherwise.*

$$x \Vdash \nabla \hat{A} \iff x \Vdash \bigwedge_{t \in U} \bigcirc_t \hat{A}(t)$$

The following construction is the multi-step limit case $U = T$.

4.2 Trajectory Logics

Definition 39 (Trajectory Coalgebra). Let $\mathbb{S} = (\mathbb{T}, S, \Phi)$ be a dynamical system. The \mathcal{I}^T -coalgebra $(S, s \mapsto \Phi_s)$ is called the *trajectory coalgebra* of \mathbb{S} .

Lemma 40. *The ∇ modality of trajectory coalgebras amounts to*

$$s \Vdash \nabla \hat{A} \iff \Phi(s, t) \Vdash \hat{A}(t) \text{ for all } t \in T$$

The ∇ trajectory modality is a surprisingly powerful logical operator, with the severe disadvantage that there is no canonical syntactic representation. The following examples are but a small subset of useful special cases.

Example. Arguments of the ∇ trajectory modality are functions of type $T \rightarrow Prop$. Various intensional notations for such functions, or time-dependent formulas, give rise to well-known temporal operators. Note that all following examples work for finitary ∇ .

- Consider discrete time $(\mathbb{N}, 0, +)$ or $(\mathbb{Z}, 0, +)$. Define a *zip* operator as

$$A \rightleftharpoons B = \nabla t \mapsto \begin{cases} A & t \text{ even} \\ B & t \text{ odd} \end{cases}$$

Then a dynamic system is bipartite, with characteristic formula A , if and only if $(A \rightleftharpoons \neg A) \vee (\neg A \rightleftharpoons A)$ is valid in the Moss model associated with its trajectories.

- Consider automaton time $(\Sigma^*, \varepsilon, \cdot)$. Define a *consumption* operator as

$$eat(L, A, B) = \nabla t \mapsto \begin{cases} A & t \in L \\ B & t \notin L \end{cases}$$

for languages $L \subseteq \Sigma^*$ and formulas A, B . Now let A be a formula characterizing accepting states. Then an automaton, as a dynamical system, accepts

- at least the language $L \subseteq \Sigma^*$ if and only if $eat(L, A, \top)$
- exactly the language $L \subseteq \Sigma^*$ if and only if $eat(L, A, \neg A)$

is valid for its initial state(s) in the Moss model associated with its trajectories.

- Consider time with a linear antisymmetric order $<$. Define a *change* operator as

$$chg(t, A, B, C) = \nabla u \mapsto \begin{cases} A & u < t \\ B & u = t \\ C & u > t \end{cases}$$

for time duration t and formulas A, B, C . Then minimum/maximum-duration operators can be defined directly, in two variants differing in the inclusion of boundary cases:

$$\begin{aligned} \min t. A &= chg(t, A, \top, \top) & \max t. A &= chg(t, \top, \top, \neg A) \\ \min' t. A &= chg(t, A, A, \top) & \max' t. A &= chg(t, \top, \neg A, \neg A) \end{aligned}$$

Imprecise operators such as *until* can be expressed as infinitary disjunctions:

$$A \mathbf{U} B = \bigvee_{t \in T} chg(t, A, B, \top)$$

4.3 Orbit Logics

The following construction shifts the coalgebraic focus from trajectories to orbits which are images of trajectories, hence abstracting from durations. The result is a family of qualitative temporal logics that can be expressed naturally in the classical modal operators, uniformly for all kinds of time structure.

Definition 41 (Orbit Coalgebra). Let $\mathbb{S} = (\mathbb{T}, S, \Phi)$ be a dynamical system. The \mathcal{P} -coalgebra (S, Φ°) is called the *orbit coalgebra* of \mathbb{S} . We say that in \mathbb{S} , y is *reachable* from x , written $x \rightsquigarrow_{\mathbb{S}} y$, if and only if $y \in \Phi^\circ(x)$.

Lemma 42. For dynamical systems \mathbb{S} , the reachability relation $\rightsquigarrow_{\mathbb{S}}$ is

1. always a preorder,
2. additionally non-branching, but not generally linear, if \mathbb{S} is linear-time,
3. additionally symmetric if \mathbb{S} is invertible.

Proof. We have $x \rightsquigarrow_{\mathbb{S}} y$ if and only if there is some t such that $\Phi(x, t) = y$. We say $x \rightsquigarrow_{\mathbb{S}} y$ via t .

1. Reflexivity and transitivity follow directly from the monoid axioms: $x \rightsquigarrow_{\mathbb{S}} x$ via 0, and if $x \rightsquigarrow_{\mathbb{S}} y$ via t and $y \rightsquigarrow_{\mathbb{S}} z$ via u , then $x \rightsquigarrow_{\mathbb{S}} z$ via $t + u$.
2. Assume that $x \rightsquigarrow_{\mathbb{S}} y$ via t and $x \rightsquigarrow_{\mathbb{S}} z$ via u . By linearity of \mathbb{T} assume, without loss of generality, that $t \leq_{\mathbb{T}} u$ via v . Then $y \rightsquigarrow_{\mathbb{S}} z$ via v .
3. For symmetric \mathbb{T} , if $x \rightsquigarrow_{\mathbb{S}} y$ via t , then $y \rightsquigarrow_{\mathbb{S}} x$ via $-t$. □

The weakening in case 2 of the preceding proposition is necessary.

Example (Nonlinear Linear-Time Dynamical System). Set $T = \{0\}$, giving rise to the singleton monoid which is trivially linear. This fixes Φ completely as $\Phi(s, t) = \Phi(s, 0) = s$, giving rise to a “still-life” structure of time. Then neither $x \rightsquigarrow_{\mathbb{S}} y$ nor $y \rightsquigarrow_{\mathbb{S}} x$ for $x \neq y$.

Definition 43 (Orbital Frame). A Kripke frame is called *orbital* if and only if it corresponds, in the sense of Lemma 31, to the orbital coalgebra of some dynamical system. An orbital frame is called *linear-time/invertible* if and only if it corresponds to the orbital coalgebra of some linear-time/invertible dynamical system, respectively.

Using the preceding definition, Lemma 42 extends to Kripke frames.

Lemma 44. *For any orbital Kripke frame $\mathbb{F} = (W, R)$, the relation R is*

1. *always a preorder,*
2. *additionally non-branching if \mathbb{F} is linear-time,*
3. *additionally symmetric if \mathbb{F} is invertible.*

This statement has a partial, finitary converse.

Lemma 45. *A finite Kripke frame (W, R) is*

1. *always orbital if R is a preorder,*
2. *additionally linear-time if R is non-branching,*
3. *additionally invertible if R is symmetric.*

Proof. Construct a dynamical system $\mathbb{S} = (\mathbb{T}, S, \Phi)$ with $(\rightsquigarrow_{\mathbb{S}}) = R$. In any case, clearly $S = W$. Proceed in reverse order and increasing flexibility of cases. For the latter two, consider the partition of W into *strongly connected components* (sccs) of the preorder R : maximal subsets $C \subseteq X$ such that $x R y$ for all $x, y \in C$. We write $x \sim y$ if and only if x, y are in the same scc, that is $x R y$ and $y R x$.

3. Set $\mathbb{T} = (\mathbb{Z}, 0, +)$. By symmetry of R there are no related pairs across sccs. For each component C choose an arbitrary cyclic permutation. Set Φ^1 to their composition. Then

- $x \rightsquigarrow_{\mathbb{S}} y$ via some $i < k$, where k is the size of the scc containing both, if $x R y$, and
- otherwise $x \not\rightsquigarrow_{\mathbb{S}} y$.

2. Set $\mathbb{T} = (\mathbb{N}, 0, +)$. We say that y is a *successor* of x , writing $x \ll y$, if and only if $x R y$ but not $y R x$. Clearly, $x R y$ if and only if either $x \sim y$ or $x \ll y$. We say that x is *transient* if it has successors. Since W is finite and R is non-branching, every transient x has a unique least successor x' , and all elements reachable from x are successors. Set $\Phi^1(x) = x'$. For non-transient x , all elements reachable from x are in the same scc. Proceed as above. Then

- $x \rightsquigarrow_{\mathbb{S}} y$ via some $i < k$, where k is the number of successors of x , if $x \ll y$,
- $x \rightsquigarrow_{\mathbb{S}} y$ via some $i < k$, where k is the size of the scc containing both, if $x \sim y$, and
- otherwise $x \not\rightsquigarrow_{\mathbb{S}} y$.

1. There are in general no least successors, and there may non-successors reachable from transient elements. A more basic construction is needed: Set $\mathbb{T} = (\mathbb{N}^*, \varepsilon, \cdot)$, the free monoid over \mathbb{N} . For each $x \in W$ choose some infinite sequence $y = (y_0, y_1, \dots) \in W^\omega$ such that $x R z$ if and only if $z = y_i$ for some i . This is always possible since the set $\{z \mid x R z\}$ is finite and nonempty. For the generating steps $\{\Phi^n \mid n \in \mathbb{N}\}$, set $\Phi^n(x) = y_n$. Then

- $x \rightsquigarrow_S y$ via 1, if $x R y$, and
- otherwise $x \not\rightsquigarrow_S y$. □

Theorem 46. *The modal logics $S4/S4.3/S5$ are sound and complete for arbitrary/linear-time/invertible orbital frames, respectively.*

Proof. $S4/S4.3/S5$ are sound for the class of Kripke frames (W, R) where R is an arbitrary/non-branching/symmetric preorder, respectively. By Lemma 44, they are also sound for the subclasses of arbitrary/linear-time/invertible orbital frames, respectively.

$S4/4.3/S5$ are complete for the class of Kripke frames (W, R) where R is an arbitrary/non-branching/symmetric preorder, respectively, and have the finite model property. By Lemma 45, they are also complete for the subclasses of arbitrary/linear-time/invertible orbital frames, respectively. □

Example. The operators \Box and \Diamond are well-suited to express “long-term” behavioral properties of dynamical systems. For instance, let A be the characteristic formula of a subset $U \subseteq S$ of the state space. Then U is a stationary solution of a dynamical system if and only if $A \rightarrow \Box A$ is valid in the Moss model associated with its orbits.

5 Conclusion

Many operators discussed in the temporal logic literature can be subsumed under a common framework by viewing them as instances of Moss’s modality ∇ , for some coalgebraic presentation of the underlying dynamical system models. As a rule of thumb,

- step coalgebras go with discrete time,
- trajectory coalgebras go with quantitative operators for either discrete or dense time, and
- orbit coalgebras go with arbitrary time and qualitative operators, in particular the classical modal operators and the framework of normal modal logics.

The examples given in this article are of course only a small selection to prove the viability of the approach. There is considerable potential for generalization. The trajectory modality is an extremely expressive tool, and it is likely that many other temporal operators can be shown to coincide with particular intensional notations for it. Besides, coalgebraic perspectives on dynamical systems other than the three detailed above could be considered. An interesting open problem and direction for future research is the integration of measure-theoretic temporal operators, for instance in duration calculus [3], into the framework.

References

- [1] G. D. Birkhoff. *Dynamical Systems*. American Mathematical Society, 1927.
- [2] M. M. Bonsangue and A. Kurz. “Duality for Logics of Transition Systems”. In V. Sassone (Ed.): *FoSSaS*. Lecture Notes in Computer Science 3441. Springer, 2005, pp. 455–469. DOI: [10.1007/978-3-540-31982-5_29](https://doi.org/10.1007/978-3-540-31982-5_29).
- [3] Z. Chaochen, C. A. R. Hoare and A. P. Raven. “A Calculus of Durations”. In: *Information Processing Letters* 40.5 (1991), pp. 269–276.
- [4] C. Cirstea et al. “Modal Logics are Coalgebraic”. In E. Gelenbe, S. Abramsy and V. Sassone (Eds.): *BCS Int. Acad. Conf.*. British Computer Society, 2008, pp. 128–140.
- [5] B. Jacobs. “Object-oriented hybrid systems of coalgebras plus monoid actions”. In: *Theoretical Computer Science* 239.1 (2000), pp. 41–95. DOI: [10.1016/S0304-3975\(99\)00213-3](https://doi.org/10.1016/S0304-3975(99)00213-3).
- [6] L. S. Moss. “Coalgebraic Logic”. In: *Ann. Pure Appl. Logic* 96.1–3 (1999), pp. 277–313. DOI: [10.1016/S0168-0072\(98\)00042-6](https://doi.org/10.1016/S0168-0072(98)00042-6).
- [7] V. Pratt. “Semantical Considerations on Floyd–Hoare Logic”. In: *Proc. 17th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 1976, pp. 109–121. DOI: [10.1109/SFCS.1976.27](https://doi.org/10.1109/SFCS.1976.27).

- [8] J. Rutten: “Universal coalgebra: a theory of systems”. In: *Theoretical Computer Science* 249.1 (2000), pp. 3–80. DOI: [10.1016/S0304-3975\(00\)00056-6](https://doi.org/10.1016/S0304-3975(00)00056-6).
- [9] Y. Venema. “Temporal Logic”. In L. Goble (Ed.): *The Blackwell Guide to Philosophical Logic*. Blackwell, 2001, Chap. 10. DOI: [10.1111/b.9780631206934.2001.00013.x](https://doi.org/10.1111/b.9780631206934.2001.00013.x).